

Evaluación de un Modelo Ontológico basado en la Adecuación Funcional de la Norma ISO 25010 para la Elicitación de Requisitos de Software

Rebeca Yuan¹, Carlos Salgado², Mario Peralta², Alberto Sánchez²

¹ Ingeniería de Software, Departamento en Ingeniería en Sistemas de Información,
Facultad Regional San Francisco – Universidad Tecnológica Nacional
San Francisco, Córdoba, Argentina
ryuan@sanfrancisco.utn.edu.ar

² Departamento de Informática,
Facultad de Ciencias Físico-Matemáticas y Naturales, Universidad Nacional de San Luis,
Ejército de los Andes 950 – C.P. 5700 – San Luis – Argentina
{csalgado, mperalta, alfanego}@unsl.edu.ar

Resumen. En la era de la transformación digital, los productos de software buscan evolucionar más allá del simple hecho de resolver un problema; deben ser capaces de apoyar con la información que generan, las decisiones de las organizaciones. Esta exigencia no proclama cambios de paradigmas para el proceso de desarrollo del software; sino una construcción “consciente” del mismo. Para lograr un juicio común de lo que se busca de un producto de software, se desarrolló un modelo ontológico como herramienta que de soporte a la elicitación de requerimientos basado en la completitud funcional de la Norma ISO 25010. El presente trabajo busca validar el modelo ontológico desarrollado, con el objetivo de establecer si el modelo resulta ser una herramienta capaz de identificar y visualizar, funciones y datos de gran impacto para las organizaciones.

Palabras Claves: Calidad de Software, Modelo Ontológico, Elicitación de Requerimientos, Adecuación Funcional, Completitud Funcional.

1 Introducción

Bajo el marco de la Cuarta Revolución Industrial, se van evidenciando cambios profundos (como toda revolución), los datos comienzan a jugar un papel importante y primordial en esta transformación digital [1].

Los sistemas de información no son ajenos a esta revolución, es de ellos mismos, donde surgen los datos para futuras tomas de decisiones. Este avance no proclama nuevos ciclos de vida de desarrollo, ni nuevas técnicas que acompañen al mismo; pero llama a hacer un alto y observar qué funciones del dominio generan información que produzca un valor agregado en la organización. Una vez más, el foco está en el análisis de requerimientos del software, etapa inicial en el proceso de desarrollo del mismo.

“Obtener los requisitos del sistema por medio de la observación de los sistemas existentes, discusión con los usuarios potenciales y proveedores, el análisis de tareas, etcétera, puede implicar el desarrollo de uno o más modelos y prototipos del sistema que ayuden al analista a comprender el sistema a especificar” [2]. El análisis de requerimientos basa su desarrollo en la actividad de elicitación de requerimientos, esta última, es la etapa de mayor interacción con el usuario. La extracción de requerimientos necesita distintas herramientas para construir y perfilar el alcance de un sistema [3]. Sobre esta tarea, se sientan los problemas y objetivos, necesidades y restricciones devenidos a requerimientos, que el analista debe saber interpretar y comunicar [4].

El tipo de lenguaje utilizado, el uso de homónimos en contextos poco definidos, son algunos de los peligros que se presentan en la etapa de elicitación de requerimientos; una herramienta que proclama y proporciona un vocabulario común son las ontologías [5]. El fin de la ontología es facilitar un entendimiento común del conocimiento a los miembros de un equipo de desarrollo de software, “una especificación formal, explícita de una conceptualización compartida” [6, 7]. Por las características enunciadas, se propone un modelo ontológico que sirva de base para la elicitación de requerimientos.

Para que el modelo sea capaz de garantizar una descripción completa del dominio que se analiza, se contempló la incorporación en el desarrollo del mismo, a la norma de calidad ISO/IEC 25010 [8]. El modelo de calidad definido por la norma se encuentra compuesto por 8 características: Adecuación Funcional, Eficiencia de Desempeño, Compatibilidad, Usabilidad, Fiabilidad, Seguridad, Mantenibilidad y Portabilidad. Una sub característica de la Adecuación Funcional, es la Completitud Funcional, grado en el cual el conjunto de funcionalidades cubre todas las tareas y los objetivos del usuario especificados [9]. El modelo bajo estudio, busca garantizar que el modelo ontológico cumpla la Completitud Funcional adaptándose a cualquier dominio que se aplique.

El desarrollo del modelo se realizó a través del Método 101, propuesto por Natalya F. Noy y Deborah L. McGuinness, [10,11]. El método establece siete pasos que guían al desarrollo de la ontología. Determinar el dominio y alcance de la ontología, reutilizar ontologías existentes, enumerar términos importantes de la ontología, definir clases y su jerarquía, definir las propiedades de esas clases, definir las facetas de ranuras y crear instancias para las mismas.

En base a lo establecido por el Método 101, el dominio y alcance de la herramienta desarrollada se corresponde a la etapa de elicitación de requerimientos de software.

Contemplar a las normas ISO en el desarrollo del modelo ontológico, permitió definir clases y relaciones representativas y adaptables a cualquier dominio. Las clases, propiedades y relaciones se establecieron para poder lograr información y un entendimiento general sobre cualquier dominio. Bajo la mirada de la norma, se gestó el establecimiento de las clases pensando, ¿para quién se desarrolla el software?, ¿quién consume la información que genera la solución?, ¿quiénes son los interesados o afectados a futuro a la solución que se propone?. El desarrollo completo del modelo se encuentra en [12].

La herramienta que permitió el desarrollo del modelo fue Protégé [13], el mismo es un editor para construir sistemas inteligentes, permite introducir el modelo ontológico, visualizar el resultado y ejecutar sobre el mismo razonadores y consultas que mejoren su estructura y entendimiento por parte de los interesados. En la Figura 1, se presenta la vista OWLViz, donde se observan las clases utilizadas en el modelo ontológico y en la Figura 2, las clases y sus relaciones, ambas vistas son generadas a través de Protégé.

Para determinar la eficiencia de la herramienta, se sometió al modelo ontológico a una evaluación triangular.

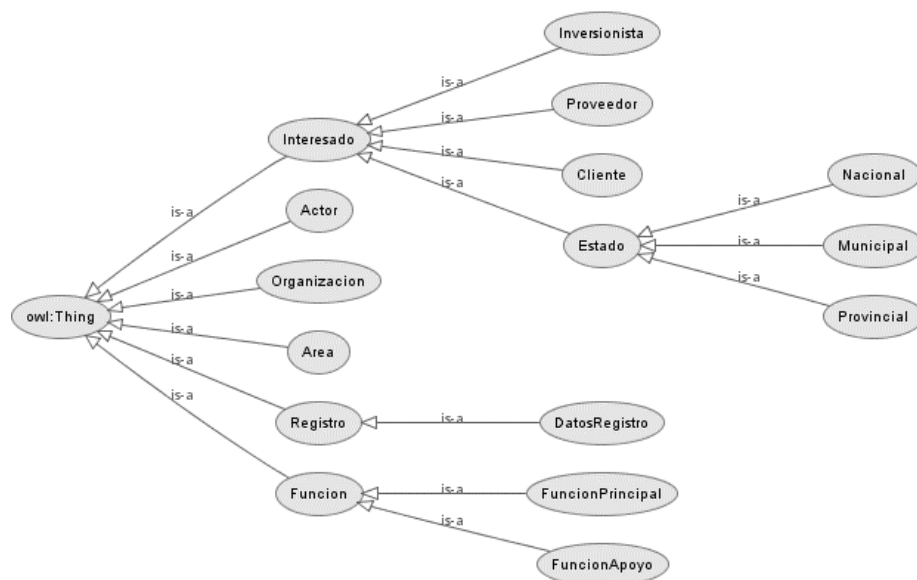


Fig. 1. Vista OWLViz Modelo Ontológico para la Elicitación de Requerimientos.

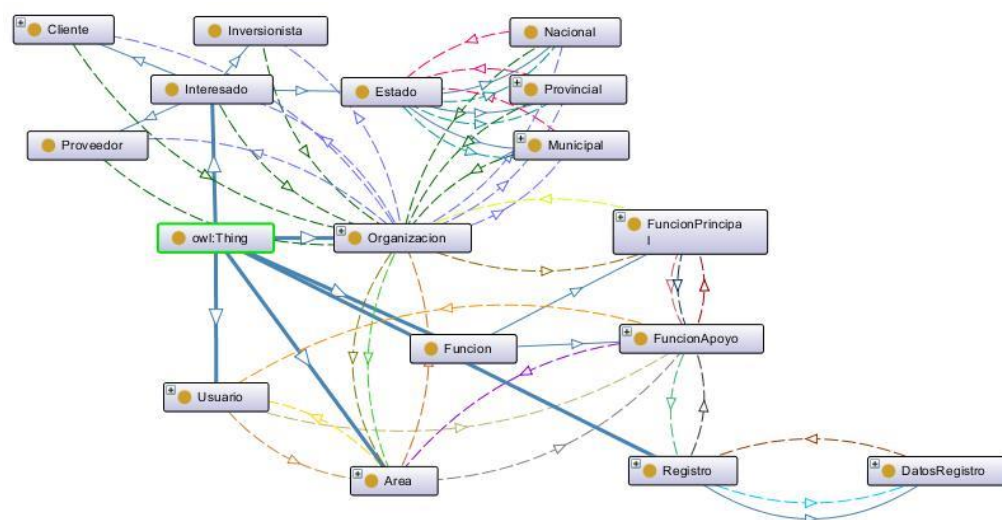


Fig. 2. Vista de Clases y Relaciones del Modelo Ontológico para la Elicitación de Requerimientos.

2 Evaluación de la Propuesta

La construcción de modelos ontológicos puede realizarse de diversas maneras. Sin embargo, a pesar de que no existe un único camino para el desarrollo de ontologías, el resultado final debe ser consecuente a los principios que lo propiciaron [14].

Algunas metodologías de desarrollo de modelos ontológicos, exponen junto a la formalización de las ontologías, el camino para evaluar las mismas [15]; otras no indican métodos de evaluación, pero plantean la necesidad de someter al modelo, a una valoración por parte de los usuarios que trabajan sobre el mismo.

La validación se refiere a que las definiciones de la ontología modelen lo más exactamente posible el dominio para el cual fueron creadas [16]. En el presente trabajo de investigación, el marco de evaluación del modelo se gestó bajo la Norma ISO/IEC 25010 [8][17] con el objetivo de que el modelo ontológico asegure la completitud funcional en la elicitación de requerimientos del software.

Para la evaluación del modelo se empleó el método de evaluación triangular propuesto en [14], el cual considera, tanto aspectos internos del modelo, como aquellos relacionados con el usuario. El método propone utilizar casos de pruebas, razonadores y una lista de chequeos con los errores más comunes detectados en el diseño de las ontologías, como se observa en la Figura 3 - “Método de evaluación triangular”. Se utilizará el razonador HermiT para evaluar las propiedades lógico-formales de las ontologías durante el ciclo de vida de las mismas. Las preguntas de competencia se realizarán a través de SPARQL. El razonador y el lenguaje SPARQL son propios de la herramienta Protégé [18], se encuentran integrados en la misma, lo que facilita la etapa de evaluación.

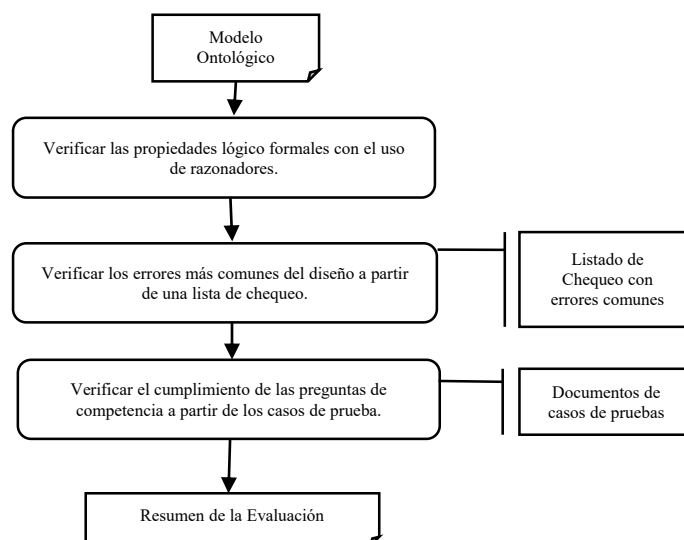


Fig. 3. Modelo de Evaluación Triangular

3 Metodología de Evaluación Triangular

3.1 Verificar las propiedades lógico formales con el uso de razonadores

La herramienta Protégé trabaja con razonadores, la función de los mismos es controlar que en el modelo establecido, no existan contradicciones e inferir en nuevo conocimiento, estos razonadores pueden estar agrupados en dos categorías: razonadores de lógica descriptiva y razonadores de programación lógica [19].

La lógica descriptiva permite representar bases de conocimiento que describen un dominio en particular, la misma se representa a través de clases, individuos, roles. Los razonadores de lógica descriptiva (LD) [19] ofrecen los siguientes servicios de inferencia: Validación de la consistencia, Validación de conceptos, Precisiones sobre los conceptos de la jerarquía.

Protégé presenta varios razonadores ELK, Ontop, Pellet y otros más, para el presente proyecto trabajamos con Hermit [20], el cual es un razonador para ontologías escritas empleando OWL (Web Ontology Language, lenguaje de marcado para publicar y compartir datos en la web) y construido empleando cálculo hypertextableau a fin de proveer razonamiento más eficiente. Pese a encontrarse integrado en Protégé, Hermit también es open-source y se anuncia como un razonador veloz y capaz de resolver ontologías complejas.

Una vez realizada la carga completa de instancias al modelo ontológico, se ejecuta el razonador. El resultado guía al analista y a los distintos actores participantes de la elicitación a visualizar aquellas características ausentes; permitiendo lograr un entendimiento compartido de las funciones que el producto de software debe realizar.

3.2 Verificar los errores más comunes del diseño a partir de una lista de chequeo.

Para verificar los errores comunes en el diseño, se toma como referencia el framework de evaluación para una bio-ontología basado en la norma de calidad ISO 9126 establecida por [21].

Los atributos a evaluar de nuestro modelo, quedan establecidos en la Tabla 1.

Tabla 1. Marco de Evaluación para el Modelo Ontológico para la Elicitación de Requerimientos de Software.

Marco de Evaluación para el Modelo Ontológico para la Elicitación de Requerimientos		
Funcionalidad	–	Referencia Ontológica
	–	Vocabulario Controlado
	–	Búsquedas y Consultas Consistentes.
	–	Adquisición de Conocimiento.
	–	Agrupamiento y similitudes.
	–	Representación de resultados.
	–	Reutilización.
	–	Inferencia.

En este contexto, la *referencia ontológica*, se corresponde con el uso de otras ontologías sobre el dominio con el que se está trabajando. Por ejemplo, si el dominio de trabajo pertenece a una empresa de Seguros, podría enlazar modelos ontológicos que

correspondan a accidentes, seguros de vida, modelos de autos, etc. El *vocabulario controlado* se establece en función de si el modelo logró extraer el vocabulario propio del dominio bajo estudio, si logra plasmar los modismos y palabras de uso común del dominio. *Adquisición de conocimiento*, se corresponde a si el modelo logra captar el entendimiento y conocimiento completo del dominio, ya sea por parte del analista, como de los usuarios y clientes; se busca determinar si el modelo sirve como herramienta que abra el debate sobre un entendimiento compartido. La característica *Representación de Resultados*, busca establecer si la herramienta presenta una interface simple para observar las instancias establecidas, bajo el mismo marco se evalúa si corresponde el *agrupamiento y similitud de las clases*, relaciones y distintos elementos del modelo. La *reutilización del modelo* para la elicitación de requerimientos en dominios diferentes va a establecer el nivel de generalidad del mismo. La *inferencia*, la *búsqueda y consultas de consistencia*, se trabajaron con el razonador Hermit.

3.3 Verificar el cumplimiento de las preguntas de competencia a partir de los casos de prueba

El último punto a evaluar corresponde a las preguntas de competencia que debe responder el modelo. Nuestro objetivo se centra en maximizar el reconocimiento de funciones del dominio; a través de la visualización por parte de los distintos actores.

Las preguntas de competencia hacen referencia a aquellos interrogantes que los actores aplican al modelo y que éste debe responder. La estructura del modelo ontológico debe encontrarse bien definida, para que permita dar respuestas verdaderas a estas preguntas.

Para realizar esta etapa de evaluación, se utilizó el Lenguaje SPARQL (Protocol and RDF Query Language/Lenguaje de Protocolo y Consulta RDF) integrado a Protégé. RDF (Por su sigla del Inglés *Resource Description Framework* - Marco de Descripción de Recursos) es un método para expresar el conocimiento en un entorno descentralizado y es el fundamento de la web semántica, en el que las aplicaciones informáticas utilizan información estructurada distribuida por toda la red [22]. RDF permite descomponer cualquier tipo de conocimiento en trozos pequeños, con reglas semánticas. SPARQL representa las RDF como un conjunto de triples que consisten en un sujeto, un predicado y un objeto como expresión básica de los datos almacenados en la base de conocimientos [18]. SPARQL brinda la posibilidad de identificar las entidades o clases que intervienen en la relación y las formas mediante las cuales se establece la relación [23]. La siguiente estructura de consulta, cubre los requisitos anteriores.

```
SELECT ?x ?y ?z
WHERE {
  ?x rdf:subClassOf ?y.
  ?y owl:onProperty ?z.
}
```

4 Casos de Estudio - Evaluación

Para realizar la evaluación del modelo ontológico; se solicitó a cuatro profesionales

en Sistemas de Información, que se desempeñan como analistas, desarrolladores y tester en distintas empresas de software, que utilicen y califiquen la herramienta.

Para calificar las variables bajo estudio, se estableció una calificación numérica que corresponde: de 1 al 3, “Regular”, de 4 a 6 calificación “Buena”; de 7 a 8 “Muy Buena” y 9 o 10 “Excelente”. Esta evaluación se realizó sobre 10 proyectos de software, de las tres empresas de software entrevistadas. Los proyectos se corresponden a aplicaciones de control, sitios web y sistemas de gestión. Los resultados obtenidos se muestran en la Tabla 2.

Tabla 2. Valoración de los distintos atributos de Funcionalidad.

		Proyecto 1	Proyecto 2	Proyecto 3	Proyecto 4	Proyecto 5	Proyecto 6	Proyecto 7	Proyecto 8	Proyecto 9	Proyecto 10
Funcionalidad	Referencia Ontológica	9	8	2	9	6	8	4	5	7	5
	Vocabulario Controlado	8	7	5	7	4	7	4	4	4	3
	Búsquedas y Consultas Consistentes.	9	7	5	8	5	6	6	6	7	6
	Adquisición de Conocimiento.	8	7	7	8	6	6	7	7	7	7
	Agrupamiento y similitudes.	7	7	6	7	7	7	7	6	5	6
	Representación de resultados.	7	7	6	7	8	8	7	8	7	8
	Reutilización.	9	8	6	7	7	6	7	7	7	7
	Inferencia.	8	8	8	6	7	7	7	6	7	7

Sobre los resultados arrojados se calculó la media y desviación estándar para analizar cada una de las características bajo estudio. La tabla 3 muestra un resumen de los resultados obtenidos en este análisis.

El análisis busca determinar si los valores establecidos por los evaluadores se corresponden a una aprobación de la herramienta, pero también es necesario establecer si existe una variación en las calificaciones, y establecer si corresponden a la herramienta o al tipo de proyecto al que se aplica.

Tabla 3. Análisis Descriptivo

	Promedio	Desviación Estándar Muestral	Máximo	Mínimo
Referencia Ontológica	6,3	2,31	9	2
Vocabulario Controlado	5,3	1,77	8	3
Búsquedas y Consultas Consistentes.	6,5	1,27	9	5
Adquisición de Conocimiento.	7	0,67	8	6
Agrupamiento y similitudes.	6,5	0,71	7	5
Representación de resultados.	7,3	0,67	8	6
Reutilización.	7,1	0,88	9	6
Inferencia.	7,1	0,74	8	6

Como se observa en la tabla 3 en los atributos Búsquedas y consultas consistentes, Adquisición de Conocimiento, Agrupamiento y similitudes, Representación de Resultados, Reutilización e Inferencia la desviación con respecto a la media es baja, su lectura se corresponde a decir que: **“de 10 proyectos analizados, representación de resultados, obtuvo una calificación media, Muy Buena, con una desviación que varía entre Bueno y Muy Bueno, por lo tanto la herramienta cumple con la expectativa de representar en forma correcta los datos que se están modelando”**.

Analizamos también aquellos puntos donde la desviación es un poco más representativa. Se realizó un análisis más exhaustivo sobre los datos de la tabla 2, para conocer qué motivó a valores tan extremos en la puntuación.

La distribución normal de la *referencia ontológica*, se corresponde a la gráfica de la figura 4.

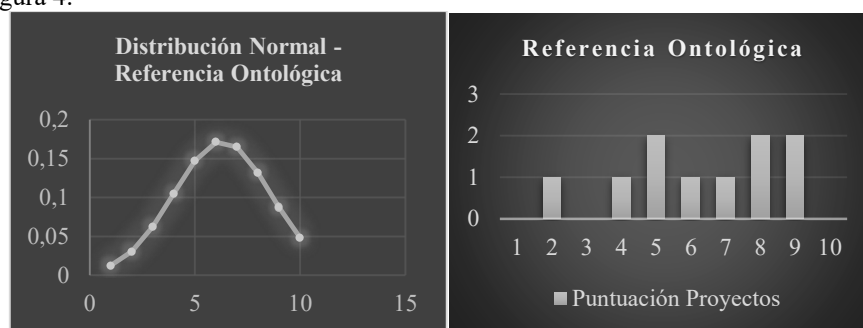


Fig. 4. Distribución Normal – Referencia Ontológica.

La gráfica muestra una distribución con una leve asimetría negativa, esto se debe a que existen en la calificación valores por debajo de la media. Analizando la distribución de los datos, se observan valores muy distantes, entre el máximo y mínimo valor, que hacen que la desviación estándar sea importante. Analizando los tipos de proyectos en los que se estableció este valor, resultó que los mismos se corresponden a un dominio muy cerrado, donde no se encontró en la web referencias ontológicas para ampliar el conocimiento del mismo. Los dominios cerrados, se corresponden a proyectos que, en su desarrollo, apuntan a solucionar problemas puntuales correspondientes a políticas de la organización.

El atributo *vocabulario controlado*, presenta una distribución asociada a la normal (Figura 5), su desviación establece que el modelo ontológico propuesto no logra extraer todo el vocabulario del dominio bajo estudio. Al analizar los proyectos asociados a valores bajos, los mismos se corresponden con proyectos de amplio alcance, es decir, dominios muy grandes.

Pese a evaluarse la herramienta en 10 proyectos, se pudo establecer dos características importantes: el *tipo* de dominio con el que se va a trabajar y el *tamaño* del mismo, infieren sobre los resultados de los atributos vocabulario y referencias, evaluados en el modelo ontológico.

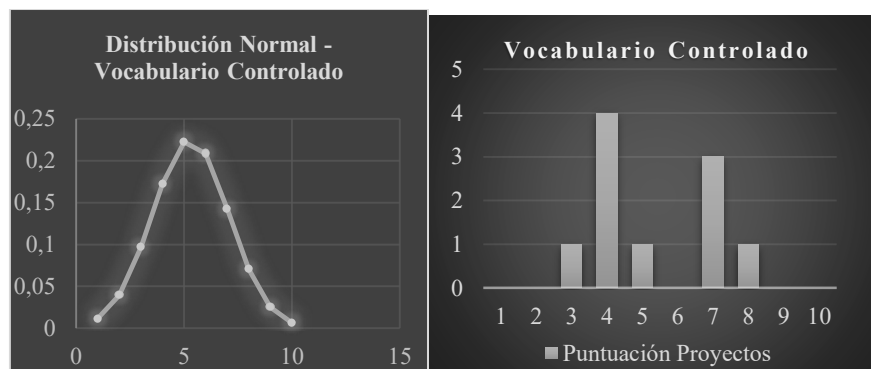


Fig. 5. Distribución Normal – Vocabulario Controlado

5 Conclusiones

La Elicitación de Requerimientos es el fundamento o base primordial en el desarrollo de proyectos software y es la fase que proporciona el impacto más alto en el diseño y en las demás fases del ciclo de vida del producto. Es muy importante asegurar la validez de los requisitos antes de comenzar el desarrollo del software.

El modelo propuesto, establece una herramienta para la elicitación de requerimientos que contribuye a solucionar el problema de completitud funcional, brindando un marco de trabajo que permite el entendimiento compartido por parte de los actores interesados en el dominio. Contemplar a la familia de normas ISO 2500 en el desarrollo y evaluación del modelo permitió visualizar la funcionalidad solicitada por los distintos actores y establecer el vocabulario del dominio bajo estudio. Contemplar a los “interesados” como clase del modelo; permitió entender el dominio como “parte de” y no atomizado al problema a resolver. Esta mirada brinda distintas perspectivas al problema y busca preguntarnos qué datos necesitamos incluir para ser competitivos a futuro. Las distinciones entre funciones principal y de apoyo, fue otro punto favorable observado, ya que indican un camino para lograr el objetivo final, y se pueden establecer planificaciones sobre las distintas funciones.

Siguiendo esta línea de investigación, sería posible aplicar herramientas de aprendizaje automático sobre la base de conocimiento que genera Protégé. Como mejoras al modelo ontológico, se observa la necesidad de incluir restricciones que hacen a la realidad de los dominios. Se concluye que evaluar al modelo ontológico bajo el marco de la norma ISO 25010 sobre adecuación funcional contribuye a la disminución de los costos de mantenimiento de software, al reducir esfuerzos y optimizar sus procesos; ya que logra obtener la comprensión y la totalidad de las funciones que los actores solicitan en la etapa de elicitación de requerimientos.

Referencias

1. Schwab, K.: La Cuarta Revolución Industrial. Penguin Random House. ISBN: 978-84-9992-

- 699-5 2016.
2. Sommerville, I.: INGENIERÍA DE SOFTWARE. Novena edición. Pearson Education. ISBN: 9786073206037. 2011.
3. M. A. Chaves, "La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software," *InterSedes Rev. las Sedes Reg.*, vol. VI, no. 10, pp. 1–13, 2005.
4. M. Griselda Báez and S. I. B. Brunner, "Metodología DoRCU para la Ingeniería de Requerimientos," pp. 210–222.
5. C. M. Z. Jaramillo, G. L. Giraldo, and G. A. U. Giraldo, "Las ontologías de la ingeniería de software: un acercamiento de dos grandes áreas del conocimiento," *Rev. Ing. Univ. Medellín*, vol. 9, no. 16, pp. 91–99, 2010, doi: 10.1017/CBO9781107415324.004.
6. G. N. Aranda and F. Ruiz, "Clasificación y ejemplos del uso de ontologías en Ingeniería del Software," *XI Congr. Argentino Ciencias la Comput.*, 2005.
7. T. R. Gruber, "A Translation Approach to Portable Ontology Specifications," 1993.
8. ISO/IEC 25010:2011 Systems and software engineering -- Systems and software Quality Requirements and Evaluation - System and software quality models. 2011.
9. J. A. M. López, "Adaptación del proceso de desarrollo software para cumplimiento de la adecuación funcional según ISO/IEC 25000," 2017.
10. Natalya F. Noy and Deborah L. McGuinness, "Ontology Development 101: A Guide to Creating Your First Ontology," *Sustain.*, vol. 9, no. 12, pp. 1–25, 2017, doi: 10.3390/su9122317.
11. O. Lassila and D. McGuinness, "The Role of Frame-Based Representation on the Semantic Web Introduction : Frame-based Representation Systems," 1992.
12. R. Yuan, C. Salgado, A. Sánchez, and M. Peralta, "Modelo Ontológico para la Educación de Requerimientos basado en la Adecuación Funcional de la Norma ISO 25010." 2019 - UTN San Francisco - Universidad Nacional de San Luis.
13. S. University, "Protégé." Available: <https://protege.stanford.edu/>. [Accessed: 10-Mar-2020].
14. Y. A. López Rodríguez, Y. Hidalgo-delgado, and N. Silega, "Un método práctico para la evaluación de ontologías," no. May, 2018.
15. M. Fernandez, A. Gómez-Pérez, and N. Juristo, "Methontology: from ontological art towards ontological engineering," *Proc. AAAI97 Spring Symp. Ser. Ontol. Eng.*, no. May 2014, pp. 33–40, 1997.
16. R. C. Esmeralda Ramos, Haydemar Núñez, "Esquema para evaluar ontologías únicas para un dominio de conocimiento," *Art War*, vol. 6, no. 3, p. 32 , ص 117, 2007, doi: 10.23943/9781400889877.
17. E. De, S. D. De, M. J. Blas, S. Gonnet, and H. Leone, "Esquema de Calidad basado en el Estándar ISO / IEC 25010 Especificación de la Calidad en Software-as-a-Service : Definición de un Esquema de Calidad basado en el Estándar ISO / IEC 25010," no. March 2018, 2016.
18. H. Dong, F. K. Hussain, and E. Chang, "Application of Protégé and SPARQL in the field of project knowledge management," *Second Int. Conf. Syst. Networks Commun. ICSNC 2007*, no. Table 1, 2007, doi: 10.1109/ICSNC.2007.22.
19. N. N. A. Helguero, "Un agente basado en un razonador de ontologías," Facultad de Ingeniería - Universidad de Buenos Aires, 2011.
20. D. A. K. G. O. C. S. O. OXFORD, "Hermit OWL Reasoner." [Online]. Available: <http://www.hermit-reasoner.com/>.
21. J. T. Fernandez-Breis, M. Egaña Aranguren, and R. Stevens, "A Quality Evaluation Framework for Bio-Ontologies," *Nat. Preced.*, no. July, 2009, doi: 10.1038/npre.2009.3479.1.
22. R. Menor, "Ricard Menor," *¿Qué es RDF?*, 2018. [Online]. Available: <https://www.seofreelance.es/que-es-rdf-introduccion-a-rdf/>.
23. A. Vega-Ramírez, I. Grangel-González, I. Sáez-Mosquera, and R. García-Castro, "Procedimiento para la obtención de un modelo ontológico para representar la información contenida en bases de datos," *CEUR Workshop Proc.*, vol. 1219, pp. 46–59, 2014.